

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of:

Confirmation No.: 4009

Dean Tan, et al.

Group Art Unit No.: 2163

Serial No.: 09/873,061

Examiner: Marie G. Cabucos

Filed: May 31, 2001

For: TECHNIQUES FOR AUTOMATICALLY
DEVELOPING A WEB SITE

MS Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed on August 15, 2008.

I. REAL PARTY IN INTEREST

Oracle International Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

The Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 54-58 have been finally rejected and are the only subjects of this appeal.

Claims 1-53 are canceled.

IV. STATUS OF AMENDMENTS

The claims were not amended after the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains one independent claim: Claim 54. Claim 54 is summarized below and annotated to cross-reference features of that claim to specific examples of those features disclosed in the specification. However, the annotations are not intended to limit the scope of the recited features to those specific examples to which the annotations refer.

Claim 54 recites (*with added reference annotations in parenthesis*) a method for building a customized web site (*page 39, lines 18-21*), wherein the method comprises performing a machine-executed operation involving instructions (*page 14, lines 13-17*), wherein the machine-executed operation is at least one of:

A) sending said instructions over transmission media (*page 17, lines 18-23; page 16, lines 11-14*);

B) receiving said instructions over transmission media (*page 17, lines 18-23; page 16, lines 11-14*);

C) storing said instructions onto a machine-readable storage medium (*page 16, lines 18-20*); and

D) executing the instructions (*page 14, lines 13-17; page 15, lines 8-14*);
wherein the instructions are instructions which, when executed by one or more processors, cause (*page 15, lines 8-14*):

storing a web site XML file (*page 46, lines 11-17*);

wherein the web site XML file is an XML document that specifies the structure of a multi-page web site (*page 70, lines 5-16—especially lines 12-13; and pages 66-68, Table 3; page 86, lines 20-22*);

wherein the web site XML file specifies (a) relationships between web pages of the multi-page web site (*page 70, lines 5-16—especially lines 14-16—and lines 20-21; and page 73, lines 20-25*), and (b) the structure and content of the pages of the multi-page web site (*page 70, lines 5-16—especially lines 13-14*);

storing XML definitions for a plurality of components that are available for use by the customized web site (*page 44, line 1—page 45, line 4—especially page 44, lines 23-26; FIG. 6C, steps 686 and 687; page 43, lines 17-22; pages 59-60, Table 2*);

presenting a user with a series of one or more user interfaces for modifying the multi-page web site to create the customized web site (*FIG. 6B, steps 632 and 640; page 41, line 22—page 42, line 7*);

wherein the one or more user interfaces include controls for adding one or more components of said plurality of components to the multi-page web site (*page 86, lines 4-9; FIG. 6E—especially add button 677b; page 84, lines 1-12; page 46, lines 11-17; page 74, lines 2-3*);

receiving through the controls user input that adds a particular component of said plurality of components to said multi-page web site (*page 46, lines 11-17; page 68, lines 10-12*);

in response to the user input, adding the XML definition associated with the particular component to the web site XML file to produce a modified web site XML file that defines a multiple-page web site that includes said particular component (*page 46, lines 11-17; page 68, lines 10-12; page 70, line 24—page 71, line 4*); and

causing a web site building component to automatically build the customized web site based on the modified web site XML file (*page 46, lines 18-25; FIG. 6B, steps 660 and 670; page 42, lines 22-25; page 74, lines 14-21; page 88, lines 12-23*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 54-56 and 58 stand rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by U.S. Patent No. 7,152,207 ("Underwood").

2. Claim 57 stands rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by Underwood.

VII. ARGUMENTS

A. The Features of Claims 54 and 55 Are Not Disclosed, Taught, or Suggested by Vervet or Nussbaum

Claim 54 recites, among other features, "storing a web site XML file; wherein the **web site XML file is an XML document that specifies the structure of a multi-page web site**; wherein **the web site XML file specifies (a) relationships between web pages of the multi-page web site**, and (b) the structure and content of the pages of the multi-page web site." Underwood does **NOT** disclose any XML document that specifies relationships between web pages of a multi-page web site.

The Examiner asserts that Underwood discloses that a definer includes various modules include a site provider/definer for defining the structure, content, and embedded applications of a web site. The Applicants respond that even if this is so, it does not mean that the structure of the web site is specified in an XML file. Furthermore, in this context, "structure of the web site" could mean several different things.

The Examiner asserts that Underwood discloses that a web definer allows a user to change the look and feel of a web site, maintains the content and navigation of the web site, without making any changes “thereto.” Apparently, this portion of Underwood is saying that, using the web definer, a user can change the look and feel of the web site without changing navigational aspects of the website. The Applicants respond that even if this is so, it does not mean that the navigational aspects of the website are stored in an XML document that specifies relationships between the website’s pages.

The Examiner asserts that Underwood discloses that the navigational links that provide access to all of the pages of a user’s website are maintained even if the user changes the look and feel of the website. This is essentially a repeat of the assertion above, and is not any more relevant to the issue of whether Underwood discloses an XML document that specifies relationships between the web pages of a web site. Even if Underwood discloses some way of maintaining navigational links between web pages of a web site, this does not in any way infer that this “way” necessarily involves an XML document that specifies navigational links, or any other relationships, between web pages of the web site. The Applicants propose that what Underwood really means is that when the visual template that spans across all of the web site’s pages is changed by a user, thus changing the “look and feel” of all of the website’s pages, the textual content that is already present on each of those web pages—including any inter-page links that might be present on those pages—is not also changed as a result of the change to the “look and feel.” Underwood does not say or mean anything more than this. Underwood does not disclose a single XML document that specifies the structure of the entire multi-page website, including the relationships between the website’s pages.

The Examiner asserts that Underwood discloses that the way that the navigational aspects of a web site's pages is maintained even throughout a change to the "look and feel" of each of the pages is via an algorithm that governs the most efficient way to maintain links between pages of the web site based on their structural relationship with one another. The Applicants concede that web pages do often have relationships with each other, and some of those relationships are structural. However, in referring to "structural relationships" here, Underwood is not referring to links between the pages, but, instead, to similarities in the structures of those pages—the structures that cause the pages to have a similar "look and feel" to each other. Regardless, even if an algorithm governs the most efficient way to maintain inter-page links during a site-wide change to the look and feel of those pages, this in no way infers the existence of some document, XML or otherwise, that specifies the relationships (interlinks or otherwise) between the web site's web pages. Clearly, there may be a variety of ways in which such an algorithm could operate. Underwood does **NOT** state that any particular way involves the XML document that is recited in Claim 55.

The Examiner asserts that Underwood discloses that fields from a "Property Page" form, with field names corresponding to property names, are automatically saved by a definer. It is not clear exactly what the Examiner is trying to prove here. Perhaps the Examiner is trying to allege that the "Property Page" is an XML document. Underwood actually says, in col. 46, lines 45-48, that the "Property Page" is "a data entry form that allows users to manipulate the properties of a DXC instance." This is quite a bit different from an XML document that specifies relationships between the pages of a multi-page website. There does not even appear to be any indication in Underwood that the "Property Page" is even an XML document. DXC instances are discussed further below because they are relevant to the understanding of what a "Property Page," discussed by the Examiner,

actually is. Although the definer might save fields from the property page form, Underwood does **NOT** disclose that these fields are saved in an XML document, or that these fields somehow specify relationships between the pages of a multi-page website.

The Examiner asserts that in col. 5, line 39, Underwood discloses "Property Page may be provided to allow the user to modify the individual elements of the XML structure." Actually, col. 5, line 39 does not say this at all. The portion of Underwood to which the Examiner refers here appears to be the portion that is actually in col. 49, lines 39-40. This portion, and the surrounding context, says:

However, Property Page may not correspond directly with the properties defined for a DXC. For example, the **entire state of the DXC may be serialized as EXtensible Markup Language ("XML")** and stored in a single property. Rather than making the user edit **the XML document** directly, a Property Page may be provided to allow the user to modify **the individual elements of the XML structure**. To achieve this, Property Page Interface 7070 and Edit Event Interface 7065 are used. Form fields that do not correspond to property names are created using Property Page Interface 7070. When the property page form is submitted, Definer forwards all of the form data to Edit Event Interface 7065, which constructs the XML document from the form data and returns it as a property update in the header of its response.

When considered in context, it is clear that the "XML document" to which the cited portion refers is nothing more than a serialized representation of the DXC (which is discussed below). A DXC is **not** a document that defines the structure of a multi-page website or indicates relationships between the pages of such a web-site. The portion quoted above is merely saying that after the DXC's state has been serialized and stored in an XML document, the user does not need to edit the raw XML document, such as by using an XML editor. Instead, the user can edit the DXC through the fields of the Property Page, which is a data entry form through whose fields the individual elements in the XML document (the one that represents the DXC). Each field of the Property Page apparently corresponds to a different XML element in the XML document that represents the DXC in serialized form.

However, the DXC, even when serialized as an XML document, does not specify the structure of an entire multi-page website, and does not indicate the relationships between the pages of such a website.

The following assertion made by the Examiner, that “Underwood recognizes XML structures of web sites’ pages exist,” does not in any way follow rationally from the text quoted above. Perhaps the foregoing quoted paragraph could be argued to demonstrate that Underwood “recognizes” that a DXC’s serialized state can be stored in an XML document, but nothing more. Regardless, the Applicants are not disputing whether or not an individual web page can be stored as an XML document. Rather, the Applicants contend that Underwood does not disclose an XML document that specifies the structure of a multi-page web site, including the relationships between the pages of that web site. These are two very different issues.

The Examiner then asserts that some of Underwood’s figures show a site map that lists pages included in a site. The Applicants admit that Underwood does, indeed, show such a site map, for whatever relevance it has to the present discussion. The Applicants even concede that site maps that illustrate the pages that are present in a website were well known. However, this does **NOT** in any way infer that, underlying the site map, there must be some XML document that somehow specifies the contents of the site map. Underwood does not disclose such an XML document. Underwood discloses other kinds of XML documents, like the unrelated-to-the-site-map one that represents the DXC, as discussed above, but not an XML document that represents a site map. Even if the site map could be stored in some persistent way, Underwood contains no teaching or suggestion that the site map’s contents would or should be stored within an XML document specifically.

The Examiner still appears to operate under a mistaken perception that a DXC specifies the structure of a multi-page website. As will be seen from the discussion below, a DXC does **NOT** specify the structure of a multi-page website.

The Examiner points to Underwood's col. 49, lines 34-50, as allegedly disclosing the features of Claim 54. This portion of Underwood does **NOT** disclose that an XML document or XML file specifies relationships between web pages of a multi-page web site. The XML document discussed in this portion of Underwood does **NOT** even specify a "hierarchy" such as the one shown in Underwood's FIG. 68.

Instead, the XML document discussed in this portion of Underwood is the serialization of a state of a "DXC" (col. 49, lines 35-38). A DXC does **NOT** specify relationships between web pages of a multi-page site. Instead, a DXC is an external web-server-hosted application that generates custom content for inclusion **within a page** of a Definer web site (col. 46, lines 15-18). A DXC is a third party **application** or component (col. 46, lines 22-24). A DXC produces HTML representations of DXC instances, which are specific instances of a DXC **embedded in a web page** (col. 46, lines 28-30). Thus, these DXC instances are nothing more than third-party applications that are embedded within web pages of a web site. DXC, and the instances thereof, do **NOT** specify the **structure** of such a web site as a whole, **NOR** do they specify **relationships between** pages of such a web site.

Indeed, it makes little sense to argue that a DXC instance, embedded within a particular page of a web site, would also specify the structure of that web site or relationships between all of the pages of that web site. There is no reason why something embedded within a single page of a web site would specify relationships between all pages of that web site. The DXC instance apparently produces content for consumption by the viewer of the

particular web page in which that DXC instance is embedded, rather than being a part of a tool that is used to generate a web site of which such a particular web page is a part.

Thus, even if a DXC can be serialized as XML, DXC has nothing to do with the specification of a multi-page web site, nor does DXC have anything to do with the specification of relationships between pages of a multi-page web site. Even if other parts of Underwood discuss a hierarchy of some kind, this hierarchy is not expressed by the DXC in any way.

Therefore, Underwood does **NOT** disclose, teach, or suggest, “storing a web site XML file; wherein the **web site XML file is an XML document that specifies the structure of a multi-page web site**; wherein **the web site XML file specifies (a) relationships between web pages of the multi-page web site**, and (b) the structure and content of the pages of the multi-page web site” as recited in Claim 54. Consequently, Claim 54 is patentable over Underwood under 35 U.S.C. § 102(e).

By virtue of their dependence from Claim 54, Claims 55, 56, and 58 inherit the features that are distinguished from Underwood above. Consequently, Claims 55, 56, and 58 also are patentable over Underwood under 35 U.S.C. § 102(e). The rejection of Claims 54-56 and 58 should be reversed.

B. The Features of Claim 57 Are Not Disclosed, Taught, or Suggested by Underwood

By virtue of its dependence from Claim 54, Claim 57 inherits the features that are distinguished from Underwood above. Therefore, for at least this reason, Claim 57 is patentable over Underwood under 32 U.S.C. § 102(e). The rejection of Claim 57 should be reversed.

Additionally, Claim 57 recites, “wherein a particular component, of the plurality of components, is dynamically generated at a different computer than a computer storing the web site XML file.” See, for example, the specification at page 42, lines 21-26; page 46, lines 18-26; page 88, lines 13-19; and page 91, lines 1-9. Although Underwood discloses a server 105 that is separate from a client terminal 125, Underwood does not disclose that any component that may be used by the web site is ever generated on any computer other than server 105. Although a user may access pages that pertain to the web site from client terminal 125, all of the component generation actually occurs on server 105.

VIII. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the rejections of Claims 54-58 lack the requisite factual and legal bases. Appellants respectfully request that the Honorable Board **reverse** the rejections of Claims 54-58.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Date: October 14, 2008

/ChristianANicholes#50,266/

Christian A. Nicholes

Registration No. 50,266

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Tel: (408) 414-1224
Fax: (408) 414-1076

CLAIMS APPENDIX

54. A method for building a customized web site, wherein the method comprises performing a machine-executed operation involving instructions, wherein the machine-executed operation is at least one of:

- A) sending said instructions over transmission media;
- B) receiving said instructions over transmission media;
- C) storing said instructions onto a machine-readable storage medium; and
- D) executing the instructions;

wherein the instructions are instructions which, when executed by one or more processors, cause:

storing a web site XML file;

wherein the web site XML file is an XML document that specifies the structure of a multi-page web site;

wherein the web site XML file specifies (a) relationships between web pages of the multi-page web site, and (b) the structure and content of the pages of the multi-page web site;

storing XML definitions for a plurality of components that are available for use by the customized web site;

presenting a user with a series of one or more user interfaces for modifying the multi-page web site to create the customized web site;

wherein the one or more user interfaces include controls for adding one or more components of said plurality of components to the multi-page web site;

receiving through the controls user input that adds a particular component of said plurality of components to said multi-page web site;
in response to the user input, adding the XML definition associated with the particular component to the web site XML file to produce a modified web site XML file that defines a multiple-page web site that includes said particular component; and
causing a web site building component to automatically build the customized web site based on the modified web site XML file.

55. The method of Claim 54, wherein causing the web site building component to automatically build the customized web site includes:
causing the web site building component to create a database for storing the customized web site.
56. The method of Claim 54, wherein execution of the instructions further cause:
creating an extensible stylesheet language transformation (XSLT) document for forming a web page; and
presenting the user with a series of one or more web pages based on the web site XML file and the XSLT document.
57. The method of Claim 54, wherein a particular component, of the plurality of components, is dynamically generated at a different computer than a computer storing the web site XML file.

58. The method of Claim 54, wherein a particular user interface, of the one or more user interfaces, includes a link to another web site for generating the particular component.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.